
# Cranite ä
# Wireless Access Controller

# FIPS 140-2
# Security Policy

# Contents

# List of Figures



# List of Tables



Revision History for Cranite Systems Wireless Access Controller

| Revised by | Date | Comments |
|---|---|---|
| M. Mancini | 5/23/02 | First draft complete |
| M. Mancini | 6/4/02 | Incorporated 5/31/02 comments from InfoGard |
| M. Mancini | 8/7/02 | Updated references to non-crypto software to software components rather than software modules; specified physical embodiment, software languages and misc updates |
| M. Mancini | 9/23/02 | Updates to reflect Infogard requests |
| M. Mancini | 12/5/02 | Updates to reflect Infogard review |
| M. Mancini | 12/6/02 | Updated list of cryptographic algorithms |
| M. Mancini | 3/10/03 | Incorporated NIST/CSE comments |

# 1.0 Introduction

This security policy defines all security rules under which the Cranite Wireless Access Controller (WAC) must operate.

The WAC is a *cryptographic software system* that operates as a multi-chip standalone cryptographic module. The cryptographic boundary of the WAC is the self-contained compiled code that is installed by the customer or reseller into production-quality compliant computer hardware. The physical boundary is the hardware platform, such as a typical PC, on which the WAC software is installed.

**Wireless Access Controller Software**
**Cryptographic Boundary**

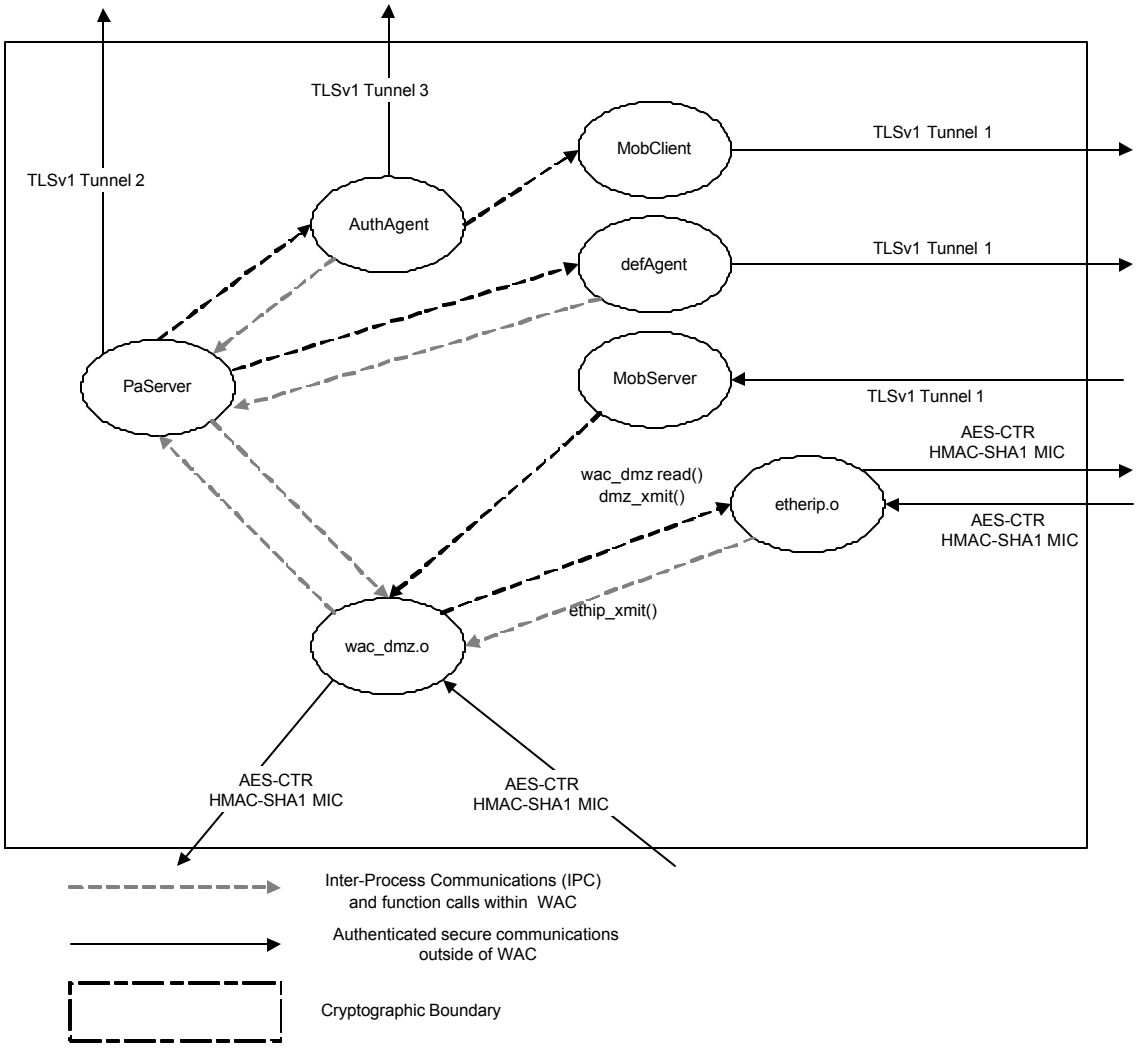


**Figure 1 - Cryptographic Boundary**

Figure 1 defines the cryptographic boundaries for the Wireless Access Controller (WAC). Contained within the cryptographic boundary are software components described in section 2.0. Each software component uses at least one CSP to perform its function. The "authenticated secure communications outside of the WAC" facilitate system functions including transfer of non-cryptographic related state information outside of the cryptographic module, secure authentication, and encrypted data communication. TLS Tunnel 1 is a single or multiple TLS tunnels that communicate from WAC to WAC. TLS Tunnel 2 is a TLS tunnel that communicates from the WAC to a Policy Server, as defined below. TLS Tunnel 3 is a TLS tunnel used for securing the mobile node authentication process, as defined below.

The WAC software and computer hardware combination operates as an *electronic encryption device*, which secures wireless networks by enabling authentication, encryption and packet filtering of networked mobile devices and access points. We use an overlay architecture to provide complete security between the wireless client device and the wired network.

The WirelessWall system, which includes the WAC as a component, delivers security through five primary architectural elements:

- Defined trust relationships that ensure no single system element can compromise the integrity of the entire system.
- An authentication framework that safeguards users' credentials regardless of the underlying system authentication mechanisms.
- Data encryption performed at Layer 2 of the Open System Interconnection model (OSI) for enhanced defense against network intrusions, denial of service attacks, and theft of data.
- Flexible security policies integrated with popular corporate directories for easy policy creation and management.
- Fine-grained packet filtering that allows authorization by server, application, protocol, or subnet.

There are three primary components that make up the WirelessWall system: the Wireless Access Controllers, the Mobile Nodes (or clients), and the Policy Server.

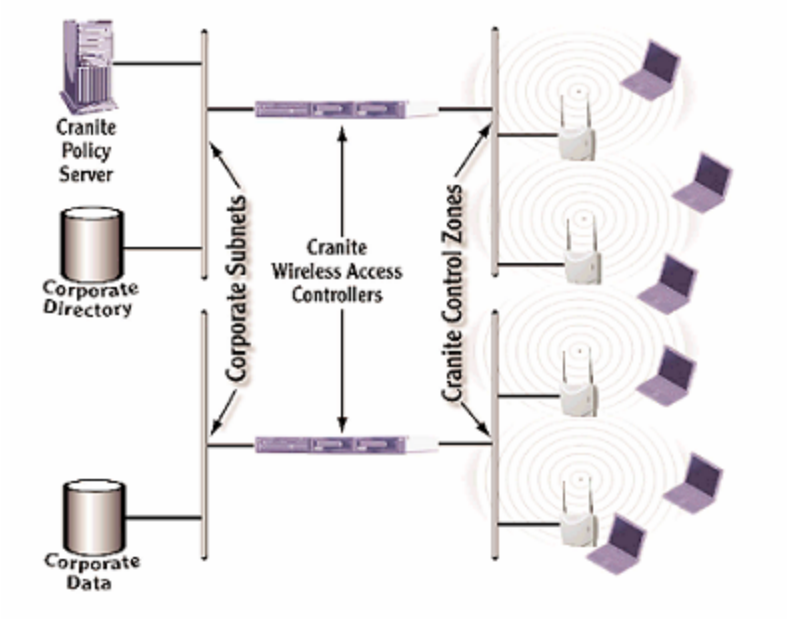


**Figure 2 - Architecture Overview**

**Wireless Access Controllers (WAC)** – Responsible for handling client authentication, AES encryption/decryption, enforcing connection policies and handling transitions for roaming users. The WAC manages connections using the 802.1x standard. It also integrates with existing enterprise authentication servers, such as RADIUS, through authentication protocols like Microsoft®-Challenge Handshake Authentication Protocol (MS-CHAP), Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) protocols. This is the module under certification.

**Client (Mobile Node – MN)** – The laptop or other device connected to the wireless network. The client contains user login and device driver code that handles authentication, state, and encryption/decryption of network traffic.

**Policy Server (PS)** – Responsible for managing policies and per-user security context information to facilitate user roaming and WAC state recovery. The PS contains a Hyper Text Markup Language (HTML) -based configuration interface to specify policies and apply policies to users and groups. The PS integrates with existing directory services on the corporate network, including LDAP, Novell® Directory Server, Microsoft® NT Domain Server and Microsoft® Active Directory.

# 2.0 Cryptographic Boundary

The WAC's hardware cryptographic boundary is the machine in which the WAC software is loaded. The WAC's software cryptographic boundary includes the following software components that perform the specified functions, as seen in Figure 1:

- **authagent** – the WAC authentication agent (AA). The AA is responsible for establishing secure communications between the WAC and a connecting MN. It also receives and sends authentication server requests and status, and contains the cryptographic algorithms for AES key establishment.
- **paServer** – the WAC provisioning agent (PA). The PA is responsible for receiving and sending security context information to and from the WAC via a secure tunnel. This tunnel communicates with entities that authenticate using the Configuration role.
- **MobServer** – the Mobility Server (MA). The MA is responsible for receiving updated WAC and MN random nonces from the Mobility Client. The random nonces are used with the master secret contained in the Security Context for AES key establishment.
- **MobClient** – the Mobility Client (MC). The MC is responsible for sending updated WAC and MN random nonces to the Mobility Server. The random nonces are used with the master secret contained in the Security Context for AES key establishment.
- **defAgent** – the Default Agent (DA). The DA is responsible for registering itself with the Policy Server via a secure tunnel.
- **wac_dmz.o** – the WAC DMZ component. The WAC DMZ component receives encryption keys and processes network traffic by encrypting and decrypting network traffic for each authenticated session.
- **etherip.o** – EtherIP component handles the encapsulation and decapsulation of Ethernet-within-IP packets (IP protocol number 97) between WACs. It is required by the WAC DMZ component. It accepts Cranite data frames to be encapsulated from the WAC DMZ component, and tunnels them to the destination WAC through normal IP routing. It also decapsulates incoming Ethernet-within-IP packets and transfers the resulting Cranite data frame to the WAC DMZ component through in-kernel mechanisms.

# 3.0 Security Level

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2. Table 1 shows the security levels for the different sections.

**Table 1 - Module Security Level Requirements**

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Cryptographic Module Ports and Interfaces | 1 |
| Roles and Services Authentication | 1 |
| Finite State Model | 1 |
| Physical Security | N/A |
| Operational Environment | 1 |
| Cryptographic Key Management | 3 |
| EMI/EMC | 3 |
| Self-Tests | 1 |
| Design Assurance | 1 |
| Mitigation of other attacks | N/A |

# 4.0 Roles and Services

## 4.1 Cryptographic Services

The following cryptographic services are available on the Wireless Access Controller (WAC):

- Receive encrypted data – this service processes encrypted data frames by authenticating each frame and decrypting it using the appropriate keys
- Send encrypted data – this service processes unencrypted data frames destined for authenticated users by encrypting the frame using the appropriate keys and sending it to the destination user
- Run self-test – this service performs cryptographic self tests on all the cryptographic algorithms and appropriate functions within the cryptographic module
- Restart System – this service restarts the cryptographic functions, including reloading the cryptographic module and running the cryptographic self-tests
- Show status – this service provides status information on the users authenticated to the system as well as various status information regarding the cryptographic module operations
- Send Security Context – this service sends security context information via a secure tunnel to the Configuration operator; this service is used to propagate certain security information to additional servers in order to facilitate secure authentication when a user roams to a new Wireless Access Controller
- Receive Security Context – this service receives security context information via a secure tunnel from the Configuration operator; this service is used to propagate a security context to additional servers in order to facilitate secure authentication when a user roams to a new Wireless Access Controller. A Security context includes the WAC-to-MN Master Secret and MN identifying information.
- Send Mobility Registration – this service sends user operator information for a mobile user, by which the key establishment protocol is executed, to operators authenticated in the Mobility role; this service is used in order to ensure incoming and outgoing network traffic is delivered to the appropriate network(s) and is both encrypted and authenticated

- Receive Mobility Registration - this service receives user operator information, by which the key establishment protocol is executed, to operators authenticated in the Mobility role; this service is used in order to ensure incoming and outgoing network traffic is delivered to the appropriate network(s) and is both encrypted and authenticated
- Authenticate User – this service authenticates User role or Cryptographic Officer operators to the cryptographic module
- Authenticate Role – this service authenticates Configuration role and Mobility role operators, which are roles that are only served to software components; these roles authenticate via certificates using the TLS protocol.
- Zeroize System – this service clears and overwrites all cryptographic keys and CSPs.

## 4.2 Operator Roles

The cryptographic module supports four distinct operator roles. These operator roles are:

- User role
- Cryptographic Officer role
- Configuration role
- Mobility role

### 4.2.1 User Role

The User role is provided only to users accessing the cryptographic services through a Mobile Node (MN). With the MN software, users may authenticate the system for the sole purpose of having access to the cryptographic services necessary for the secure transport of data over an insecure network.

### 4.2.2 Cryptographic Officer Role

The role is available for the purpose of restarting the cryptographic functions and executing various administrative services.

### 4.2.3 Configuration Role

The Configuration role is served to software components communicating via a secure channel with the WAC following a successful authentication using the TLS protocol. The Configuration role uses the Send Security Context and Receive Security Context services following the successful authentication of User role operators and following the expiration of User role operator sessions.

### 4.2.4 Mobility Role

The Mobility role is served to software components, specifically the Mobility Server and Mobility Client software, for the purpose of transmitting and receiving certain Critical Security Parameters needed to facilitate securing the network connections for mobile, or roaming, operators authenticated to the User role.

## 4.3 Services Available to Each Role

Table 2 lists the services available to each role.

**Table 2 - Services and Roles**

| Cryptographic Service | User Role | Cryptographic Officer Role | Configuration Role | Mobility Role |
|---|---|---|---|---|
| Receive encrypted data | X | | | |
| Send encrypted data | X | | | |
| Run self-test | | X | | |
| Restart System | | X | | |
| Show status | | X | | |
| Send Security Context | | | X | |
| Receive Security Context | | | X | |
| Send Mobility Registration | | | | X |
| Receive Mobility Registration | | | | X |
| Authenticate User | X | X | | |
| Authenticate Role | | | X | X |
| Zeroize System | | X | | |

## 4.4 Authentication Methods for Each Operator Role

### 4.4.1 User Role

The cryptographic module authenticates operators in performing the User role via identity-based authentication. This authentication occurs through client software, which communicates via a secure tunnel with the Wireless Access Controller. The User role is granted access to the cryptographic services through the user software following a successful authentication to the cryptographic module. The authentication occurs using the secure tunnel via a challenge sent to the operator. The operator challenge response is sent to the WAC, which then passes the challenge response to an appropriate authentication server. Success or failure of the authentication requests is determined based on the response from the authentication server. If the User role authentication request is successful, the user's policy is retrieved from a separate policy server. Provided the user, based on their identity (username and password), has a policy, the User role is authenticated for that operator and certain cryptographic services become available to the operator.

### 4.4.2 Cryptographic Officer Role

The Cryptographic Officer role is available only to an authenticated administrator who performs authentication on the Wireless Access Controller (WAC). This authentication occurs using the WAC operating system login process.

### 4.4.3 Configuration Role

The Configuration role is authenticated using certificates signed by a trusted CA that resides both on the WAC and the computer system containing the Configuration role software. The authentication is accomplished using the TLSv1 protocol and the certificates to ensure mutual authentication.

### 4.4.4 Mobility Role

The Mobility role is authenticated using certificates signed by a trusted CA that resides both on the WAC and the computer system containing the Mobility role software. The authentication is accomplished using the TLSv1 protocol and the certificates to ensure mutual authentication.

## 4.5 Roles and Required Identification and Authentication

**Table 3 - Roles and Authentication**

| Role | Type of Authentication | Authentication Data |
|---|---|---|
| User Role | MS-CHAP, CHAP or PAP via secure, mutually authenticated TLS tunnel based on RSA public/private certificates | Username plus up to a 128-character password |
| Cryptographic Officer Role | WAC operating system login | Root user plus password |
| Configuration Role | TLS tunnel with RSA public/private certificates for mutual authentication | TLS authentication process with signed public certificate |
| Mobility Role | TLS tunnel with RSA public/private certificates for mutual authentication | TLS authentication process with signed public certificate |

## 4.6 Strengths of Authentication Mechanisms

**Table 4 - Authentication Mechanism Strength**

| Authentication Mechanism | Strength of Mechanism | Strength against one minute attack |
|---|---|---|
| User Role Password Challenge Authentication | Better than 1 in 2^96 | Better than 1 in 2^96 |
| Operating System root login authentication | Better than 1 in 62^59 | Better than 1 in (62^59)/60 |
| Configuration and Mobility Role Authentication | Better than 1 in 2^204 | Better than 1 in 2^204 |

# 5.0 Access Control Policy

## 5.1 Critical Security Parameters

The following are the critical security parameters:

- WAC-to-MN TLS Master Secret: This CSP is used in the TLS key establishment protocol of the WAC-to-MN TLS session keys as well as the WAC-to-MN AES keys. There is a separate WAC-to-MN TLS Master Secret for each operator in the User Role
- WAC-to-MN TLS Session Keys: These keys are used to protect authentication-specific network traffic between the WAC and the MN
- WAC-to-MN AES Keys: These keys are used by the WAC to protect data transmitted between the WAC and the MN.
- WAC-to-PS TLS session keys: These keys are used to protect data transmitted between the WAC and the PS, including the TLS session contexts for each MN.
- WAC-to-WAC TLS session keys: These keys are used to protect mobility information transmitted between the WACs.
- WAC Software Key: this SHA-1-HMAC key is used during systems startup or reload to verify the integrity of the WAC software before it is loaded into memory and executed; it is also used to encrypt and decrypt the Security Context information which is written to persistent storage.
- PRNG State: the PRNG is used during key establishment of the WAC-to-MN AES keys.
- User Role Authentication Challenge Response Value: This value is the challenge response received from an operator attempting to authenticate the User role
- Cryptographic Officer Role Authentication / Credentials: This value is the username and password for the Cryptographic Officer role operator who can only authenticate on the WAC. These credentials are secured and managed by the WAC operating system
- Private Key used for TLS Authentication: This RSA private key is used to mutually authenticate the WAC to operators accessing various cryptographic services.

Each TLS Session Key set includes two (2) three-key triple DES keys (one for transmit, one for receive) and two (2) SHA-1-HMAC keys (one for transmit, one for receive).

The WAC-to-MN AES key set includes two (2) 128-bit AES keys (one for transmit, one for receive) and two (2) SHA-1-HMAC keys (one for transmit, one for receive).

## 5.2 Cryptographic Service Input Output Role Matrix

**Table 5 - Cryptographic Service Input Output Role Matrix**

| | | | | | Roles | | | |
|---|---|---|---|---|---|---|---|---|
| **Cryptographic Service** | **Purpose/Function** | **Input** | **Output** | **Status** | **User** | **Crypto Officer** | **Configuration** | **Mobility** |
| Receive encrypted data | Process encrypted data frames by authenticating each frame and decrypting it using the appropriate keys | Encrypted frames | Unencrypted frames | None | **X** | | | |
| Send encrypted data | Processes unencrypted data frames destined for authenticated users by encrypting the frame using the appropriate keys and sending it to the destination user. | Decrypted frames | Encrypted frames | None | **X** | | | |
| Run self-test | Performs cryptographic self tests on all the cryptographic algorithms and appropriate functions within the cryptographic module. | None | Console output indicating result of self-test | Self-test results to Log files and console | | **X** | | |
| Restart System | Restarts the cryptographic functions, including reloading the cryptographic module and running the cryptographic self-tests. | None | Console output indicating restart and startup status | Console output indicating restart and startup status | | **X** | | |
| Show status | Provides status information on the users authenticated to the system as well as various status information regarding the cryptographic module operations. | None | Console output indicating status of software components and users | Console output indicating status of software components and users | | **X** | | |
| Send Security Context | Sends security context information via a secure tunnel to the Configuration operator; this service is used to propagate certain security information to additional servers in order to facilitate secure authentication when a user roams to a new Wireless Access Controller. | User Authentication Success Message | Security Context Message | None | | | **X** | |
| Receive Security Context | Receives security context information via a secure tunnel from the Configuration operator; this service is used to propagate certain security information to additional servers in order to facilitate secure authentication when a user roams to a new Wireless Access Controller. | Security Context Message | Security Context written to disk | None | | | **X** | |

| Cryptographic Service | Purpose/Function | Input | Output | Status | User | Crypto Officer | Configuration | Mobility |
|---|---|---|---|---|---|---|---|---|
| Send Mobility Registration | Sends user operator information for a mobile user, by which the key establishment protocol is executed, to operators authenticated in the Mobility role; this service is used in order to ensure incoming and outgoing network traffic is delivered to the appropriate network(s) and is both encrypted and authenticated through the TLSv1 protocol. | - TLS abbrv handshake with remote home WAC<br>- user auth with remote home WAC | Mobility Registration Message | None | | | | X |
| Receive Mobility Registration | Receives user operator information, by which the key establishment protocol is executed, to operators authenticated in the Mobility role; this service is used in order to ensure incoming and outgoing network traffic is delivered to the appropriate network(s) and is both encrypted and authenticated through the TLSv1 protocol. | Mobility Registration Message | Mobility Registration data written to disk | None | | | | X |
| Authenticate User | Authenticates User role operators to the cryptographic module. | Username and password received through console for Crypto Officer; through TLS tunnel for User role | Login succeeded or Login failed | Log entry representing success or failure of Authentication request | X | X | | |
| Authenticate Role | Authenticates Configuration role and Mobility role operators, which are roles that are only served to software components; these roles authenticate via certificates using the TLS protocol. | TLS Handshake | TLS Handshake | None | | | X | X |
| Zeroize System | Clears and overwrites all cryptographic keys. | Zeroize command | Console output indicating when Zeroize is completed | Console output indicating when Zeroize is completed | | X | | |

## 5.3 Critical Security Parameter Access Based On Role

**Table 6 - Critical Security Parameter Access Based on Role**

| Critical Security Parameter | User Role | Cryptographic Officer Role | Configuration Role | Mobility Role |
|---|---|---|---|---|
| WAC-to-MN TLS Master Secret | x | x | x | x |
| WAC-to-MN TLS Session Keys | x | x | | |
| WAC-to-MN AES Keys | x | x | | |
| WAC-to-PS TLS session keys | | x | x | |
| WAC-to-WAC TLS session keys | | x | | x |
| WAC Software Key | | x | x | |
| PRNG State | x | x | x | x |
| User Role Authentication Challenge Response Value | x | | | |
| Cryptographic Officer Role Authentication / Credentials | | x | | |
| Private key used for TLS Authentication | x | x | x | x |

## 5.4 Access Rights Within Services

**Table 7 - Access Rights Within Services**

| Cryptographic Service | WAC-to-MN TLS Master Secret | WAC-to-MN TLS Session Keys | WAC-to-MN AES Keys | WAC-to-PS TLS session keys | WAC-to-WAC TLS session keys | WAC Software Key | PRNG State | User Role Authentication Challenge Response | Cryptographic officer Role Authentication / | Private TLS Key |
|---|---|---|---|---|---|---|---|---|---|---|
| Receive encrypted data | | | r | | | | | | | |
| Send encrypted data | | | r | | | | | | | |
| Run self-test | | | | | | r | rw | | | |
| Restart System | | | | rw | rw | | | | | |
| Show status | | | | | | | | | | |
| Send Security Context | r | | | r | | r | | | | r |
| Receive Security Context | w | | | r | | r | | | | r |
| Send Mobility Registration | | | | | r | | | | | |
| Receive Mobility Registration | | | | | r | | | | | |
| Authenticate User | rw | rw | rw | r | | | r | r | r | r |
| Authenticate Role | | | | | r | | r | | | r |
| Zeroize System | z | z | z | z | z | z | z | z | | z |

**r = read, w = write, z = zeroize**

## 5.5 Modes of Access to Critical Security Parameters

**Table 8 - Modes Of Access to Critical Security Parameters**

| Role | Critical Security Parameter | Use |
|---|---|---|
| User | WAC-to-MN TLS Master Secret | Used in the key establishment of the WAC-to-MN TLS Session Keys as well as the WAC-to-MN AES keys. |
| User | WAC-to-MN TLS Session Keys | Used to encrypt TLS communications during the User role authentication process. |
| User | WAC-to-MN AES Keys | Used to encrypt and decrypt messages data traffic sent between the WAC and MN |
| User | PRNG State | Used to provide entropy when computing the WAC to MN AES Keys. |
| User | User Role Authentication Challenge Response Value | Provided by User role for the purpose of responding to an authentication challenge. |
| User | Private key used for TLS Authentication | Used by the User role to authenticate the WAC; used to establish a TLS connection between the MN and WAC. |

| Role | Critical Security Parameter | Use |
|---|---|---|
| Cryptographic Officer | WAC-to-MN TLS Session Keys | Can be zeroized by executing the zeroization service or reset by restarting the WAC. |
| Cryptographic Officer | WAC-to-MN AES Keys | Can be zeroized by executing the zeroization service or reset by restarting the WAC. |
| Cryptographic Officer | WAC-to-PS TLS Session Keys | Can be zeroized by executing the zeroization service or reset by restarting the WAC. |
| Cryptographic Officer | WAC-to-WAC TLS Session Keys | Can be zeroized by executing the zeroization service or reset by restarting the WAC. |
| Cryptographic Officer | WAC Software Key | Can be read or zeroized. Zeroizing this key will prevent the WAC from loading the cryptographic module. |
| Cryptographic Officer | Cryptographic Officer Role Authentication / Credentials | Used to authenticate the WAC operating system for the purpose of having access to the Cryptographic Officer role services; the credentials may be changed using the OS's change password command. |
| CryptographicOfficer | Private key used for TLS Authentication | Can be read, manually changed or zeroized. |

| Role | Critical Security Parameter | Use |
|---|---|---|
| Configuration | WAC-to-MN TLS Master Secret | Used to communicate minimal security information to WACs in order to facilitate abbreviated TLS authentication when a MN connects to a new WAC. |
| Configuration | WAC-to-PS TLS Session Keys | Used to transmit the TLS Master Secret CSP via a secure TLS tunnel between the WAC and PS. |
| Configuration | WAC Software Key | Used to encrypt and decrypt the Security Context information which contains the WAC-to-MN TLS Master Secret and is stored in encrypted format in the WAC's persistent storage. |
| Configuration | Private key used for TLS Authentication | Used to establish a mutually authenticated TLS tunnel between the WAC and PS and establish the WAC-to-PS TLS Session Keys. |

| Role | Critical Security Parameter | Use |
|---|---|---|
| Mobility | WAC-to-WAC TLS Session Keys | Used to transmit and receive the Server and Client Random Nonces from one WAC to another WAC for an authenticated mobile user (roaming MN). |
| Mobility | PRNG State | Used to provide entropy when determining a new set of AES keys. |
| Mobility | Private key used for TLS Authentication | Used to establish a mutually authenticated TLS tunnel between WACs and establish the WAC-to-WAC TLS Session Keys. |

# 6.0 General Rules

The FIPS-version of the Wireless Access Controller has only one mode of operations – FIPS mode. When the WAC is started, this mode of operation is entered automatically with no operator intervention. To enable the FIPS mode of operation, simple start up the WAC.

## 6.1 Access Control Prior to Cryptographic Module Initialization

After the WAC software is installed, but before the cryptographic module is initialized, there are no cryptographic services available to operators other than the Cryptographic Officer role. The Cryptographic Officer role is available to allow that operator to configure the WAC software and load the cryptographic module. The WAC Management Manual specifies how to configure and initialize the cryptographic module.

## 6.2 Concurrent Operators

The cryptographic module does not support multiple operator roles. The operating system does not allow concurrent operator access.

## 6.3 Software Security.

The WAC software is written in C and operates on the Linux operating system. The software is installed in the host hardware storage medium as compiled binary executable components. At system initialization and restart, the cryptographic module software components are each authenticated using HMAC and the WAC Software Key to ensure the software has not been modified. If the Software Load Test passes, the unencrypted software is loaded into the system memory, if it fails, the cryptographic module is placed in the Crypto Failure state and can only exit that state following a successful Software Load Test.

## 6.4 Operating System Security

The WAC operates automatically after power-up. OS functions are only available to an operator authenticating using the Cryptographic Officer role.

## 6.5 Protection of Authentication Data.

During operator authentication, passwords are masked from entry and are not echoed to the operator console.

## 6.6 Procedures for Zeroizing the System.

The cryptographic officer may Zeroize the cryptographic module by authenticating the WAC and typing the following command:

```
zeroize –all
```

The operator will then be prompted to confirm their request. If confirmed, the cryptographic system will shutdown and all CSPs will be zeroized.

## 6.7 Cryptographic Self-Tests

Self-tests are initiated with the cryptographic module is loaded. Each module within the software cryptographic boundary that contains cryptographic algorithms initiates the cryptographic self-test at load time. Table 9 lists the tests performed by modules when loading.

**Table 9 - Module Tests**

| Component | Cryptographic Algorithm | CypherSuite/Mode | Test Type |
|---|---|---|---|
| AuthAgent | RSA | | Known Answer Test |
| | TDES | TDES_EDE_CBC_SHA | Known Answer Test |
| | PRNG | | Known Answer Test |
| PaServer | RSA | | Known Answer Test |
| | TDES | TDES_EDE_CBC_SHA | Known Answer Test |
| | PRNG | | Known Answer Test |
| MobServer | RSA | | Known Answer Test |
| | TDES | TDES_EDE_CBC_SHA | Known Answer Test |
| | PRNG | | Known Answer Test |
| MobClient | RSA | | Known Answer Test |
| | TDES | TDES_EDE_CBC_SHA | Known Answer Test |
| | PRNG | | Known Answer Test |
| defAgent | RSA | | Known Answer Test |
| | TDES | TDES_EDE_CBC_SHA | Known Answer Test |
| | PRNG | | Known Answer Test |
| wac_dmz.o | AES | CTR | Known Answer Test |
| | HMAC-SHA1 | | Known Answer Test |
| etherip.o | n/a | n/a | n/a |

The results of all self tests are output to the system log files stored in the `/var/log` directory. If any self-test fails, an error message will be sent to the Cryptographic Officer console and the system will disable all cryptographic functions by unloading the cryptographic module and placing the Cryptographic Module in the Crypto Failure State. Additionally, error messages will be written to the cryptographic system log files located in the `/var/log` directory.

To initiate a self-test the cryptographic officer must first authenticate him/herself to the WAC and then type the following commands:

```
service wac-agents stop
service bridge restart
service wac-agents start
```

## 6.8 Continuous PRNG Test

The FIPS-approved PRNG used in the Cryptographic Module executes a continuous PRNG test which ensures that each call to the random number generator yields a different result from the previously generated 160-bit block. If the same 160-bit block is returned, the Cryptographic Module is placed in the Crypto Failure State.

### 6.9 Determining the Status of the Cryptographic Module

In order to determine the status of the cryptographic module, the cryptographic officer must first authenticate him/herself to the WAC and then type the following commands:

**`service wac-agents status`**
[This will display the status of the various cryptographic module software components; each should indicate "running" as their state.]
**`service bridge status`**
[This will display the status of the data encryption / decryption cryptographic module; it should display a list of MAC addresses that the component sees on its network interfaces.]

The cryptographic officer can also view the status of the cryptographic module by viewing the log files stored in the `/var/log` directory.

### 6.10 Error State Handling

Should any cryptographic processing encounter an error condition that places the cryptographic module in the Cryptographic Failure State, the error condition will be written to the log files and the cryptographic module will be disabled by the cryptographic module being unloaded from system memory. The only way to recover from the Error State is by re-running the Self-tests, and thus re-initiating the system, as specified above.

### 6.11 Re-Authentication Process following Power Cycle

Following Power Cycle any operator authenticated in the Cryptographic Officer role must reauthenticate to the Cryptographic Module as though that user had never been authenticated. All User role operators authenticate the Cryptographic Module using an abbreviated TLS handshake. No User role operator is granted access to the Cryptographic Module, under this circumstance, unless the abbreviated TLS handshake is completed successfully. Following a successful TLS handshake, new WAC-to-MN AES keys are created. All other roles must complete a full authentication process as specified for each role.

## 7.0 Physical Security Policy

The WAC software is installed by the customer or VAR on production-quality, FCC-certified hardware (such as a PC), which also defines the module's physical boundary.

## 8.0 Mitigation of Other Attacks

The cryptographic module is not designed to mitigate attacks outside the scope of FIPS 140-2.

## 9.0 Cryptographic Algorithms Used

The following cryptographic algorithms are used:

FIPS Approved:

- FIPS-approved Pseudo-Random Number Generator (PRNG) - Implemented FIPS 186-2 appendix 3.1 PRNG
- HMAC-SHA1
- RSA PKCS#1
- AES-ECB

- AES-CTR
- AES-CBC
- TDES-TCBC
- SHA-1

Non-FIPS Approved:

- RSA public key cipher (key establishment)
- MD5 (key establishment)

## 10.0 Acronym List

| | |
|---|---|
| AA | Authentication Agent |
| AES | Advanced Encryption Standard |
| CA | Certificate Authority |
| CBC | Cipher Block Change |
| CHAP | Challenge Handshake Authentication Protocol |
| CSP | Crypto Security Parameter |
| CTR | Counter Mode |
| DA | Default Agent |
| DMZ | Demilitarized Zone |
| ECB | Electronic Code Block |
| EDE | Encryption Decryption Encryption |
| FCC | Federal Communications Commission |
| FIPS | Federal Information Processing Standards |
| HMAC | Keyed-Hashing for Message Authentication |
| HTML | Hypertext Markup Language |
| IP | Internet Protocol |
| IPC | Inter-Process Communication |
| LDAP | Lightweight Directory Access Protocol |
| MA | Mobility Agent |
| MAC | Medium Access Control |
| MC | Mobility Client |
| MD5 | Message-Digest Algorithm |
| MIC | Message Integrity Check |
| MN | Mobile Node |
| MS-CHAP | Microsoft Challenge Handshake Authentication Protocol |
| OS | Operating System |
| OSI | Open Systems Interconnection |
| PA | Provisioning Agent |
| PAP | Password Authentication Protocol |
| PC | Personal Computer |
| PRNG | Psuedo Random Number Generator |
| PS | Policy Server |
| RADIUS | Remote Authentication Dial-In User Service |
| RSA | Rivest-Shamir-Adelman |
| SHA-1 | Secure Hash Algorithm |
| TCBC | TDEA Cipher Block Chaining Mode of Operation |
| TDEA | Triple Data Encryption Algorithm |
| TDES | Triple Data Encryption Standard |
| TLS | Transport Layer Security |

| | |
|---|---|
| VAR | Value Added Reseller |
| WAC | Wireless Access Controller |